



# Toward the first quantum simulation with quantum speedup

Andrew M. Childs<sup>a,b,c,1</sup>, Dmitri Maslov<sup>b,c,d</sup>, Yunseong Nam<sup>b,c,e</sup>, Neil J. Ross<sup>b,c,f</sup>, and Yuan Su<sup>a,b,c</sup>

<sup>a</sup>Department of Computer Science, University of Maryland, College Park, MD 20742; <sup>b</sup>Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742; <sup>c</sup>Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, MD 20742; <sup>d</sup>Division of Computing and Communication Foundations, National Science Foundation, Alexandria, VA 22314; <sup>e</sup>IonQ, Inc., College Park, MD 20740; and <sup>f</sup>Department of Mathematics and Statistics, Dalhousie University, Halifax, NS B3H 4R2, Canada

Edited by John Preskill, California Institute of Technology, Pasadena, CA, and approved August 10, 2018 (received for review January 30, 2018)

**With quantum computers of significant size now on the horizon, we should understand how to best exploit their initially limited abilities. To this end, we aim to identify a practical problem that is beyond the reach of current classical computers, but that requires the fewest resources for a quantum computer. We consider quantum simulation of spin systems, which could be applied to understand condensed matter phenomena. We synthesize explicit circuits for three leading quantum simulation algorithms, using diverse techniques to tighten error bounds and optimize circuit implementations. Quantum signal processing appears to be preferred among algorithms with rigorous performance guarantees, whereas higher-order product formulas prevail if empirical error estimates suffice. Our circuits are orders of magnitude smaller than those for the simplest classically infeasible instances of factoring and quantum chemistry, bringing practical quantum computation closer to reality.**

quantum computing | quantum simulation | quantum circuits

**W**hile a scalable quantum computer remains a long-term goal, recent experimental progress suggests that devices capable of outperforming classical computers will soon be available (refs. 1–5; [www.research.ibm.com/ibm-q/](http://www.research.ibm.com/ibm-q/)). Multiple groups have already developed programmable devices with several qubits and two-qubit gate fidelities around 98% (6), and similar devices with around 50 qubits are under active development. While the error rates of these early machines severely limit the total number of gates that can be reliably performed, future improvements should lead to machines with more qubits and more-reliable gates. This raises the exciting possibility of solving practical problems that are beyond the reach of classical computation. Such an outcome would be a landmark in the development of quantum computers and would begin an era in which they serve not only as test beds for science but as practical computing machines.

Reaching this goal will require not only significant experimental advances but also careful quantum algorithm design and implementation. Here we address the latter issue by developing explicit circuits, and thereby producing concrete resource estimates, for practical quantum computations that can outperform classical computers. Through this work, we aim to identify applications for small quantum computers that help to motivate the significant investment required to develop scalable, fault-tolerant quantum computers.

There has been considerable previous research on compiling quantum algorithms into explicit circuits (see *SI Appendix, section A* for more detail). However, to the best of our knowledge, none of these studies aimed to identify minimal examples of superclassical quantum computation, and typical resource counts were high. Our work is also distinct from recent work on quantum computational supremacy (7), where the goal is merely to accomplish a superclassical task, regardless of its practicality. Instead, we aim to pave the way toward practical quantum computations (which may not be far beyond the threshold for supremacy).

Arguably, the most natural application of quantum computers is to the problem of simulating quantum dynamics (8). Quantum computers can simulate a wide variety of quantum systems, including fermionic lattice models (9), quantum chemistry (10), and quantum field theories (11). However, simulations of spin systems with local interactions likely have less overhead, so we focus on them as an early candidate for practical quantum simulation. Note that analog quantum simulation (4, 5) is another promising approach to simulating spin systems that may be easier to realize in the short term. However, analog simulators lack universal control, and it can be challenging to ensure correctness of their output. Here we focus on digital simulation for its greater flexibility, for the prospect of invoking fault tolerance, and for its role as a stepping-stone to other forms of quantum computation.

Efficient quantum algorithms for simulating quantum dynamics have been known for over two decades (12). Recent work has led to algorithms with significantly improved asymptotic performance as a function of various parameters such as the evolution time and the allowed simulation error (13–17). Our work investigates whether these alternative algorithms can be advantageous for simulations of relatively small systems, and aims to lay the groundwork for an early practical application of quantum computers.

## 1. Target System

To produce concrete benchmarks, we focus on a specific simulation task. Specifically, we consider a one-dimensional

### Significance

**Near-term quantum computers will have limited numbers of qubits and will only be able to reliably perform limited numbers of gates. Therefore, it is crucial to identify applications of quantum processors that use the fewest possible resources. We argue that simulating the time evolution of spin systems is a classically hard problem of practical interest that is among the easiest to address with early quantum devices. We develop optimized implementations and perform detailed resource analyses for several leading quantum algorithms for this problem. By evaluating the best approaches to small-scale quantum simulation, we provide a detailed blueprint for what could be an early practical application of quantum computers.**

Author contributions: A.M.C., D.M., Y.N., N.J.R., and Y.S. designed research, performed research, and wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

Published under the [PNAS license](https://www.pnas.org/licenses).

Data deposition: The implementations of quantum algorithms for the simulation of Hamiltonian dynamics in the Quipper quantum programming language have been deposited on GitHub and are available at <https://github.com/njross/simcount>.

<sup>1</sup>To whom correspondence should be addressed: Email: [amchilds@umd.edu](mailto:amchilds@umd.edu).

This article contains supporting information online at [www.pnas.org/lookup/suppl/doi:10.1073/pnas.1801723115/-DCSupplemental](http://www.pnas.org/lookup/suppl/doi:10.1073/pnas.1801723115/-DCSupplemental).

Published online September 6, 2018.

nearest-neighbor Heisenberg model with a random magnetic field in the  $z$  direction. This model is described by the Hamiltonian

$$\sum_{j=1}^n (\vec{\sigma}_j \cdot \vec{\sigma}_{j+1} + h_j \sigma_j^z), \quad [1]$$

where  $\vec{\sigma}_j = (\sigma_j^x, \sigma_j^y, \sigma_j^z)$  denotes a vector of Pauli  $x$ ,  $y$ , and  $z$  matrices on qubit  $j$ . We impose periodic boundary conditions (i.e.,  $\vec{\sigma}_{n+1} = \vec{\sigma}_1$ ), and  $h_j \in [-h, h]$  is chosen uniformly at random. The parameter  $h$  characterizes the strength of the disorder.

This Hamiltonian has been considered in recent studies of self-thermalization and many-body localization (see *SI Appendix, section B* for more detail). Despite intensive investigation, the details of a transition between thermal and localized phases remain poorly understood. A major challenge is the difficulty of simulating quantum systems with classical computers; indeed, the most extensive numerical study we are aware of was restricted to, at most, 22 spins (18).

Hamiltonian simulation can efficiently access any feature that could be observed experimentally (and more), and there are several proposals for exploring self-thermalization by simulating dynamics (19–21). Since all of these approaches involve only very simple state preparations and measurements, we focus on the cost of simulating dynamics. We consider evolution times comparable to the number of spins, since the system must evolve for this long for self-thermalization to take place (or even for information to propagate across the system, owing to the Lieb–Robinson bound).

Specifically, we produce gate counts for simulations with  $h = 1$ , evolution time  $t = n$  (the number of spins in the system), and overall accuracy  $\epsilon = 10^{-3}$ . These explicit choices help us to focus on the system-size dependence of quantum simulation algorithms. This is a key consideration for practical applications, yet it has been deemphasized in the literature on sparse Hamiltonian simulation.

## 2. Implementations

There are many distinct quantum algorithms for Hamiltonian simulation, some of which are summarized in Table 1. We implement algorithms based on high-order product formulas (PFs) (13), direct application of the Taylor series (TS) (15), and the recent quantum signal processing method (QSP) (16), all of which are introduced in *SI Appendix, section C*. We expect these to be among the most efficient approaches to digital quantum simulation. In particular, approaches based on quantum walk (14, 23) appear to incur greater overhead (as discussed in *SI Appendix, section D*).

To produce concrete circuits, we implement quantum simulation algorithms in a quantum circuit description language called

Quipper (24) (see *SI Appendix, section E* for more details). Wherever possible, we tighten the analysis of algorithm parameters and manually optimize the implementation. We also process all circuits using an automated tool we developed for large-scale quantum circuit optimization (25). Our implementation is available in a public repository (<https://github.com/njross/simcount>).

We express our circuits over the set of two-qubit CNOT gates, single-qubit Clifford gates, and single-qubit  $z$  rotations  $R_z(\theta) := \exp(-i\sigma^z\theta/2)$  for  $\theta \in \mathbb{R}$ . Such gates can be directly implemented at the physical level with both trapped ions (2) and superconducting circuits (1, 2). In both technologies, two-qubit gates take longer to perform and incur more error than single-qubit gates. Thus, the CNOT count is a useful figure of merit for assessing the cost of physical-level circuits on a universal device. We also produce Clifford+ $T$  circuits using optimal circuit synthesis (26) so that we can count  $T$  gates, which are typically the most expensive gates for fault-tolerant computation.

Our analysis ignores many practical details such as architectural constraints, instead aiming to give a broad overview of potential implementation costs that can be refined for specific systems. When counting qubits, we assume that measured ancillas can be reused later.

**PF Algorithm.** The PF approach approximates the exponential of a sum of operators by a product of exponentials of the individual operators. The asymptotic complexity of this approach can be improved with higher-order Suzuki formulas (27). By splitting the evolution into  $r$  segments and making  $r$  sufficiently large, we can ensure that the simulation is arbitrarily precise. The main challenge in making these algorithms concrete is to choose an explicit  $r$  that ensures some desired upper bound on the error. *SI Appendix, section F* gives a detailed description of these implementation details.

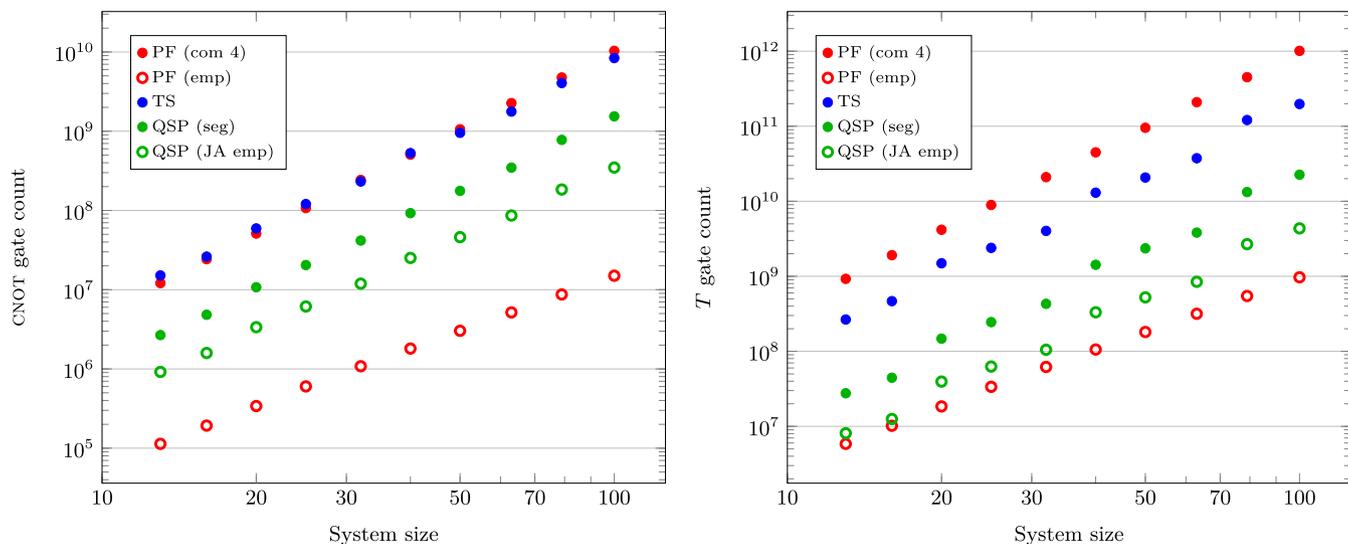
We present two bounds, which we call the “analytic” and “minimized” bounds, that slightly strengthen previous analysis (13). However, bounds of this type are far from tight (28–30). Thus, we develop an improved bound that exploits commutation relations among terms in the target Hamiltonian. For a one-dimensional system of  $n$  spins with nearest-neighbor couplings, evolving for time  $t = n$ , this commutator bound improves the asymptotic complexity of the  $(2k)$ th-order PF algorithm from  $O(n^{3+1/k})$  to  $O(n^{3+2/(2k+1)})$  while also significantly improving the leading constant.

Naive computation of the commutator bound takes time  $O(n^{2k+1})$ , which can be prohibitive even for small  $n$ . To make this approach practical, we develop techniques that exploit the structure of the Hamiltonian to compute the commutator bound in closed form. We explicitly evaluate this bound for the first-, second-, and fourth-order product formulas.

**Table 1. Gate complexities of quantum simulation algorithms**

Algorithm	Ref.	Gate complexity ( $t, \epsilon$ )	Gate complexity ( $n$ )
PF, first order	(12)	$O(t^2/\epsilon)$	$O(n^5)$
PF, $(2k)$ th order	(13)	$O(5^{2k} t^{1+1/2k} / \epsilon^{1/2k})$	$O(5^{2k} n^{3+1/k})$
Quantum walk	(14)	$O(t/\sqrt{\epsilon})$	$O(n^4 \log n)$
Fractional-query simulation	(22)	$O\left(t \frac{\log^2(t/\epsilon)}{\log \log(t/\epsilon)}\right)$	$O\left(n^4 \frac{\log n}{\log \log n}\right)$
TS	(15)	$O\left(t \frac{\log^2(t/\epsilon)}{\log \log(t/\epsilon)}\right)$	$O\left(n^3 \frac{\log^2 n}{\log \log n}\right)$
Linear combination of quantum walk steps	(23)	$O\left(t \frac{\log^{3.5}(t/\epsilon)}{\log \log(t/\epsilon)}\right)$	$O\left(n^4 \frac{\log n}{\log \log n}\right)$
QSP	(16)	$O(t + \log(1/\epsilon))$	$O(n^3 \log n)$

This table shows previously asymptotic gate complexities of quantum simulation algorithms as a function of the simulation time  $t$ , allowed error  $\epsilon$ , and the system size  $n$  for a one-dimensional nearest-neighbor spin model as in Eq. 1 with  $t = n$  and fixed  $\epsilon$ .



**Fig. 1.** Gate counts for optimized implementations of the PF algorithm (using the fourth-order formula with commutator bound and the better of the fourth- or sixth-order formulas with empirical error bound and the nonsegmented version with empirical Jacobi–Anger error bound) for system sizes between 10 and 100. (Left) CNOT gates for Clifford+ $R_z$  circuits. (Right)  $T$  gates for Clifford+ $T$  circuits.

Unfortunately, even the commutator bound can be very loose. To address this, we report empirical error estimates by extrapolating the error seen in direct classical simulations of small instances [as also explored in previous work on simulating many-body dynamics (28) and quantum chemistry (29, 30)]. While these empirical bounds do not provide rigorous guarantees on the simulation error, they may nevertheless be useful in practice, and they improve the cost of PF algorithms by several orders of magnitude.

**TS Algorithm.** The TS algorithm directly implements the (truncated) Taylor series of the evolution operator for a carefully chosen constant time using a procedure for implementing linear combinations of unitary operations (15). This segment is then simply repeated until the entire evolution time has been simulated. The circuit for a segment is built using three subroutines: a state preparation procedure, a reflection about the  $|0\rangle$  state, and an operation denoted  $\text{select}(V)$  (discussed further below). Our implementation of the TS algorithm (described in detail in *SI Appendix, section G*) also includes a concrete error analysis that establishes rigorous, non-asymptotic bounds on the simulation parameters.

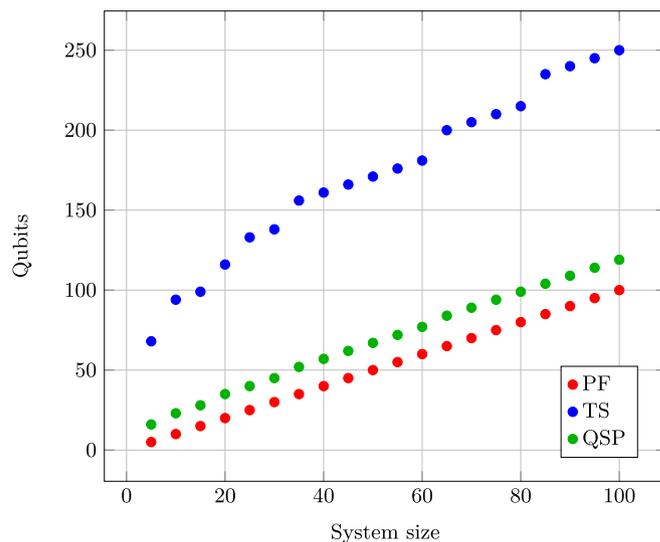
The aforementioned  $\text{select}(V)$  operation applies a unitary  $V_j$  conditioned on a control register being in the state  $|j\rangle$ , for  $j \in \{1, \dots, \Gamma\}$ . We develop an improved implementation of this operation by designing an optimized walk on a binary tree, saving a factor of about  $\log_2 \Gamma$  in the gate count. For our simulations of systems with 10 to 100 spins, this reduces CNOT and  $T$  gate counts over a naive implementation by a factor of between 5 and 9, significantly improving the overall complexity. Furthermore, the cost of our  $\text{select}(V)$  implementation meets a previously established asymptotic lower bound (31). This improvement may be more broadly applied to any algorithm using the  $\text{select}(V)$  procedure, such as others based on linear combinations of unitaries.

**QSP Algorithm.** The QSP algorithm of Low and Chuang (16, 17) effectively implements a linear combination of unitary operators by a different mechanism. This algorithm applies a sequence of operations called “phased iterates” that manifest each eigenvalue of the Hamiltonian as a rotation acting on an ancilla qubit.

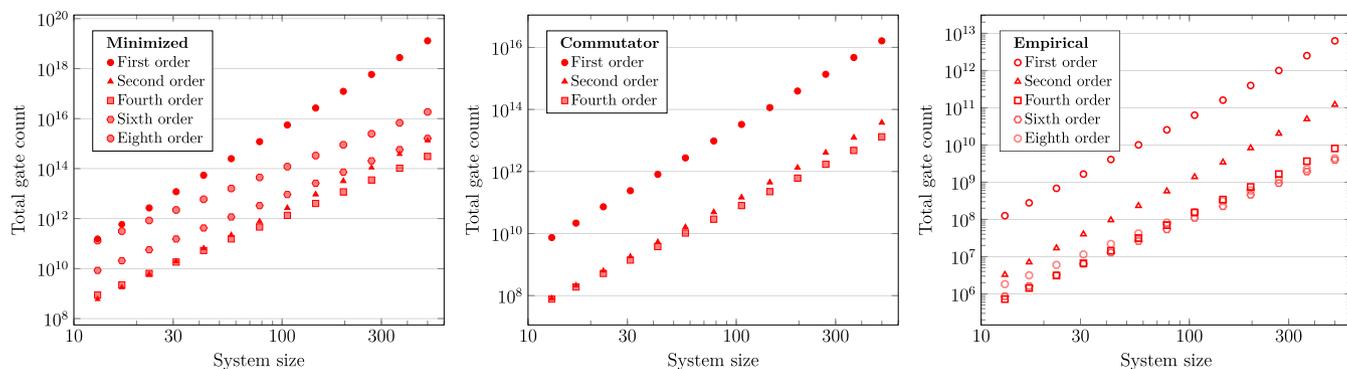
By carefully choosing a sequence of rotation angles for that qubit, we induce the desired evolution.

The circuit for each phased iterate is built from subroutines similar to the TS algorithm. However, computing the  $M$  rotation angles for the phased iterates requires finding the roots of a polynomial of degree  $2M$ , and these roots must be determined to high precision. Because of these challenges, we were unable to compute the parameters of the algorithm explicitly except in very small instances. Instead, we produced estimates of the gate count (but not a complete implementation) by synthesizing a version of the algorithm with placeholder values of the parameters.

One way to alleviate this problem is to consider a segmented implementation of the QSP algorithm. In this approach, we divide the evolution time into  $r$  segments, each of which is



**Fig. 2.** Numbers of qubits used by the PF, TS, and QSP algorithms. See *SI Appendix, Eqs. 184 and 195* for the precise qubit counts of the TS and QSP algorithms, respectively. Jumps in the qubit count of the TS algorithm correspond to increases in the size of each of  $K$  registers used to index one of  $L$  terms in the Hamiltonian, as detailed in *SI Appendix, section G*.



**Fig. 3.** Total gate counts in the Clifford+ $R_z$  basis for product formula algorithms using the (Left) minimized, (Center) commutator, and (Right) empirical bounds, for system sizes between 13 and 500.

sufficiently short that the classical preprocessing is tractable. Since the optimality of the QSP approach to Hamiltonian simulation relies essentially on simulating the entire evolution as a single segment, the segmented approach has higher asymptotic complexity. However, it allows us to develop a complete implementation, and the overhead for moderate values of  $n$  is not too high.

For the full version of the algorithm, we consider an empirical error bound on the Jacobi–Anger expansion, giving a modest improvement. Numerical evidence suggests that the additional savings from an empirical error bound for the overall algorithm would not be significant. For the segmented version of the algorithm, we instead used an analytic error bound so that the algorithm remains rigorous (and because an empirical Jacobi–Anger error bound did not give much improvement in that case).

*SI Appendix, section H* discusses our implementation of QSP algorithms in detail.

### 3. Results

Fig. 1 compares gate counts for the PF algorithm (with commutator and empirical error bounds), the TS algorithm, and the QSP algorithm (in both segmented and nonsegmented versions). The TS algorithm uses significantly more qubits than the QSP algorithm (as shown in Fig. 2) while also requiring more gates, so the latter is clearly preferred. In contrast, the QSP algorithm has only slightly greater space requirements than the PF algorithm.

Surprisingly, despite being more involved, the QSP algorithm outperforms the rigorously bounded PF algorithm even for small system sizes. In particular, among the rigorously analyzed algorithms, the segmented QSP algorithm has the best performance, improving over the PF algorithm by about an order of magnitude for CNOT count and by almost two orders of magnitude for  $T$  count.

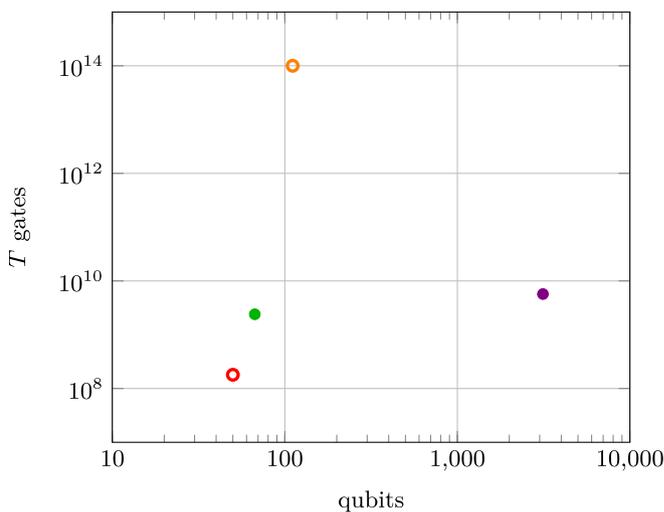
Empirical error bounds improve the performance of the PF algorithm by two to three orders of magnitude, making it the preferred approach if rigorous performance guarantees are not required. For the CNOT count, the empirical PF algorithm improves over the full QSP algorithm by about an order of magnitude. The advantage in the  $T$  count is less significant, but still indicates that the PF algorithm is dominant, especially considering its lower qubit count.

Although we expected that higher-order product formulas would not be advantageous for small system sizes, we find that the fourth- and sixth-order formulas had the best performance for our benchmark system with tens to hundreds of qubits, as shown in Fig. 3. The fourth-order formula with commutator bound gives the best available PF algorithm with a rigorous performance guarantee. Using empirical error bounds, the sixth-

order formula outperforms the fourth-order formula for systems of about 30 or more qubits, making the former the method of choice for simulations just beyond the reach of classical computers (again, provided a heuristic error bound can be tolerated). These results suggest that higher-order formulas may be advantageous for other quantum simulations, such as those for quantum chemistry, even though they have not usually been considered (30, 32).

For a system of 50 qubits—which is presumably close to the limits of direct classical simulation for circuits such as ours (33)—the TS algorithm uses 171 qubits and the QSP algorithm uses 67, whereas the PF algorithm uses only 50. [Recent work has demonstrated simulation of 56-qubit computations, but only for circuits of much smaller depth than those considered in our work (34).] At this size, the segmented QSP algorithm is the best rigorously analyzed approach, using about  $1.8 \times 10^8$  CNOT gates (over the set of Clifford+ $R_z$  gates) and  $2.4 \times 10^9$   $T$  gates (over the set of Clifford+ $T$  gates). Using the empirical error bound, the PF algorithm uses about  $3 \times 10^6$  CNOTs and  $1.8 \times 10^8$   $T$ s (over Clifford+ $R_z$  and Clifford+ $T$ , respectively).

For comparison, previous estimates of gate counts for factoring, discrete logarithms, and quantum chemistry simulations are significantly larger, as illustrated in Fig. 4. First, consider factoring a 1,024-bit number, which is beyond the factorization



**Fig. 4.** Comparison of resource requirements for factoring a 1,024-bit number (35) (purple), simulation of FeMoco (30) (orange), and 50-spin simulations described in this paper (segmented QSP in green; sixth-order PF with empirical error bound in red).

of RSA-768 that was achieved classically in 2009 (36). The best implementation we are aware of uses 3,132 qubits and about  $5.7 \times 10^9$   $T$  gates (when realized over the set of Clifford+ $T$  gates) (35). (Ref. 35 does not give explicit resource counts; we estimate them as described in *SI Appendix, section A*.) Quantum algorithms for classically hard instances of the elliptic curve discrete logarithm problem have roughly comparable cost (37). For quantum chemistry, a natural target for a problem just beyond the reach of classical computing is a simulation of FeMoco, the primary cofactor of nitrogenase, an enzyme that catalyzes the nitrogen fixation process. Even for a fairly low-precision simulation, and using nonrigorous estimates of the product formula error, the best implementation we are aware of uses 111 qubits and  $1.0 \times 10^{14}$   $T$  gates (30). Thus, it appears that simulation of spin systems is indeed a significantly easier task for near-term practical quantum computation.

For a more detailed discussion of the results, see *SI Appendix, section I*.

#### 4. Discussion

The work described in this paper represents progress toward a possible early application of quantum computers, solving a practical problem that is beyond the reach of classical computation. Our optimized implementations can perform a 50-spin quantum simulation using fewer qubits, and dramatically fewer gates (by over five orders of magnitude in the case of quantum chemistry), than comparable instances of other classically hard problems. Of course, our results only represent upper bounds. While we attempted to optimize the implementation wherever possible, it is likely that further improvements can be found, and it is conceivable that another algorithm (or computational task) may offer better performance. Our work establishes a concrete set of benchmarks that we hope can be improved through future studies.

Demonstrations of digital quantum simulation performed to date (38–41) have been limited in scope, primarily using the first-order formula [except for some limited applications of the second-order formula (38, 39)]. Our results show that higher-order formulas are useful even for simulations of small systems. In the near term, it could be fruitful to demonstrate the utility of these formulas experimentally. Even relatively small experiments might be able to probe the validity of our empirical error bounds.

We have also identified some avenues for future improvement of quantum simulation algorithms. We saw that rigorous error bounds for product formulas are very loose, even with our newly developed commutator bound. This motivates attempting to prove stronger rigorous error bounds for product formulas. Also, the difficulty of computing the angles needed to perform the QSP algorithm prevents us from taking full advantage of the algorithm in practice, so it would be useful to develop a more efficient classical procedure for specifying these angles.

After the initial version of this paper was posted, Haah et al. (42) developed a new quantum algorithm for simulating the dynamics of spatially local Hamiltonians. Their approach uses bounds on the propagation of information in a local system to give a simulation with complexity only linear in the product of the system size and the evolution time (up to log factors), asymptotically improving over all previous methods. While the gate count estimates presented in figure 3 of ref. 42 suggest that, for the example considered in this paper, the PF and QSP methods remain favorable for systems with up to about 100 spins, the techniques developed in ref. 42 are likely to be useful in relatively near-term simulations.

Further reduction of the gate count could be especially significant if it led to a simulation with sufficiently few gates to be performed without invoking fault tolerance. With our current estimate of millions of CNOT gates for a superclassical simulation, this is likely out of reach at present. However, further improvement could obviate the need for error correction in a system with highly accurate gates, making an early demonstration of superclassical simulation more accessible.

Finally, our work has considered an idealized system. Practical devices will come with architectural constraints, may use different basic operations than those considered here, may allow parallelization of gates, and will likely require fault tolerance. By incorporating such features, we hope the work begun here will lead to a blueprint for an early practical quantum computation.

**ACKNOWLEDGMENTS.** We thank Zhexuan Gong, Alexey Gorbakov, Guang Hao Low, Chris Monroe, and Nathan Wiebe for helpful discussions. This work was supported, in part, by the Army Research Office (Multidisciplinary University Research Initiative Award W911NF-16-1-0349), the Canadian Institute for Advanced Research, and the National Science Foundation (Grant 1526380). This material was partially based on work supported by the National Science Foundation during D.M.'s assignment at the foundation. Any opinion, finding, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

- Chen Y, et al. (2014) Qubit architecture with high coherence and fast tunable coupling. *Phys Rev Lett* 113:220502.
- Debnath S, et al. (2016) Demonstration of a small programmable quantum computer with atomic qubits. *Nature* 536:63–66.
- Song C, et al. (2017) 10-qubit entanglement and parallel logic operations with a superconducting circuit. *Phys Rev Lett* 119:180511.
- Bernien H, et al. (2017) Probing many-body dynamics on a 51-atom quantum simulator. *Nature* 551:579–584.
- Zhang J, et al. (2017) Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator. *Nature* 551:601–604.
- Linke NM, et al. (2017) Experimental comparison of two quantum computing architectures. *Proc Natl Acad Sci USA* 114:3305–3310.
- Harrow AW, Montanaro A (2017) Quantum computational supremacy. *Nature* 549:203–209.
- Feynman RP (1982) Simulating physics with computers. *Int J Theor Phys* 21:467–488.
- Wecker D, et al. (2015) Solving strongly correlated electron models on a quantum computer. *Phys Rev A* 92:062318.
- Wecker D, Bauer B, Clark BK, Hastings MB, Troyer M (2014) Gate count estimates for performing quantum chemistry on small quantum computers. *Phys Rev A* 90:022305.
- Jordan SP, Lee KSM, Preskill J (2012) Quantum algorithms for quantum field theories. *Science* 336:1130–1133.
- Lloyd S (1996) Universal quantum simulators. *Science* 273:1073–1078.
- Berry DW, Ahokas G, Cleve R, Sanders BC (2007) Efficient quantum algorithms for simulating sparse Hamiltonians. *Commun Math Phys* 270:359–371.
- Berry DW, Childs AM (2012) Black-box Hamiltonian simulation and unitary implementation. *Quantum Inf Comput* 12:29–62.
- Berry DW, Childs AM, Cleve R, Kothari R, Somma RD (2015) Simulating Hamiltonian dynamics with a truncated Taylor series. *Phys Rev Lett* 114:090502.
- Low GH, Chuang IL (2016) Hamiltonian simulation by qubitization. arXiv:1610.06546.
- Low GH, Chuang IL (2017) Optimal Hamiltonian simulation by quantum signal processing. *Phys Rev Lett* 118:010501.
- Luitz DJ, Laflorencie N, Alet F (2015) Many-body localization edge in the random-field Heisenberg chain. *Phys Rev B* 91:081103.
- Serbyn M, et al. (2014) Interferometric probes of many-body localization. *Phys Rev Lett* 113:147204.
- Schreiber M, et al. (2015) Observation of many-body localization of interacting fermions in a quasirandom optical lattice. *Science* 349:842–845.
- Smith J, et al. (2016) Many-body localization in a quantum simulator with programmable random disorder. *Nat Phys* 12:907–911.
- Berry DW, Childs AM, Cleve R, Kothari R, Somma RD (2014) Exponential improvement in precision for simulating sparse Hamiltonians. *Proceedings of the 46th ACM Symposium on Theory of Computing (Assoc Comput Machinery, New York)*, pp 283–292.
- Berry DW, Childs AM, Kothari R (2015) Hamiltonian simulation with nearly optimal dependence on all parameters. *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science (Inst Electr Electron Eng, New York)*, pp 792–809.
- Green AS, Lumsdaine PL, Ross NJ, Selinger P, Valiron B (2013) Qipper: A scalable quantum programming language. *ACM SIGPLAN Notices* 48:333–342.
- Nam Y, Ross NJ, Su Y, Childs AM, Maslov D (2018) Automated optimization of large-scale quantum circuits with continuous parameters. *npj Quantum Inf* 4:23.
- Ross NJ, Selinger P (2016) Optimal ancilla-free Clifford+ $T$  approximation of  $z$ -rotations. *Quantum Inf Comput* 16:901–953.
- Suzuki M (1991) General theory of fractal path integrals with applications to many-body theories and statistical physics. *J Math Phys* 32:400–407.
- Raeisi S, Wiebe N, Sanders BC (2012) Quantum-circuit design for efficient simulations of many-body quantum dynamics. *New J Phys* 14:103017.

29. Babbush R, McClean J, Wecker D, Aspuru-Guzik A, Wiebe N (2015) Chemical basis of Trotter-Suzuki errors in quantum chemistry simulation. *Phys Rev A* 91:022311.
30. Reiher M, Wiebe N, Svore KM, Wecker D, Troyer M (2017) Elucidating reaction mechanisms on quantum computers. *Proc Natl Acad Sci USA* 114:7555–7560.
31. Maslov D (2016) Optimal and asymptotically optimal NCT reversible circuits by the gate types. *Quantum Inf Comput* 16:1096–1112.
32. Poulin D, et al. (2015) The Trotter step size required for accurate quantum simulation of quantum chemistry. *Quantum Inf Comput* 15:361–384.
33. Häner T, Steiger DS (2017) 0.5 petabyte simulation of a 45-qubit quantum circuit. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (Assoc Comput Machinery, New York), p 33.
34. Pednault E, et al. (2017) Breaking the 49-qubit barrier in the simulation of quantum circuits. arXiv:1710.05867.
35. Kutin SA (2006) Shor's algorithm on a nearest-neighbor machine. arXiv:quant-ph/0609001.
36. Kleinjung T, et al. (2010) Factorization of a 768-bit RSA modulus. *Proceedings of the 30th Annual Cryptology Conference (CRYPTO)* (Springer, Berlin), pp 333–350.
37. Roetteler M, Naehrig M, Svore KM, Lauter K (2017) Quantum resource estimates for computing elliptic curve discrete logarithms. *Proceedings of the 23rd International Conference on the Theory and Applications of Cryptology and Information (ASIACRYPT 2017)* (Springer, New York), pp 241–270.
38. Brown KR, Clark RJ, Chuang IL (2006) Limitations of quantum simulation examined by simulating a pairing Hamiltonian using nuclear magnetic resonance. *Phys Rev Lett* 97:050504. arXiv:quant-ph/0601021.
39. Lanyon BP, et al. (2011) Universal digital quantum simulation with trapped ions. *Science* 334:57–61.
40. Barends R, et al. (2015) Digital quantum simulation of fermionic models with a superconducting circuit. *Nat Commun* 6:7654.
41. O'Malley PJJ, et al. (2016) Scalable quantum simulation of molecular energies. *Phys Rev X* 6:031007.
42. Haah J, Hastings MB, Kothari R, Low GH (2018) Quantum algorithm for simulating real time evolution of lattice Hamiltonians. arXiv:1801.03922.